CLAIMS

What is claimed is:

1. A method of program execution in a data processing system comprising:

means for fetching a first instruction located at a first address; and

means for executing the first instruction with the data processing system, the method comprising:

executing the first instruction;

selecting a jump address based upon a value by providing a second address for the jump address if a comparison of the value with a predetermined value has a first result, and providing a third address for the jump address if the comparison of the value with the predetermined value has a second result, wherein neither the second address nor the third address is contiguous to the first address; and

always implementing a change of control in the program execution in response to executing the first instruction by redirecting program execution to the jump address.

2. The method of claim 1 wherein the first result is a comparison determination that the value is greater than the predetermined value and the second result is a comparison determination that the value is less than or equal to the predetermined value.

3. The method of claim 1 further comprising:

implementing the value as a count value stored in a counter.

4. The method of claim 3 wherein implementing the value as a count value in a counter further comprises:

using the count value to track when a change of program tasks should be implemented by the data processing system.

5. The method of claim 1 further comprising:

generating the jump address by maintaining a program counter having an address value which gets incremented during program execution and truncating a predetermined number of low order bits of the address value and combining a resultant with an offset to create the jump address.

6. The method of claim 1 further comprising:

selecting the second address and the third address to be within a predetermined range of addresses that is less than a total range of addresses within the data processing system.

7. The method of claim 1 further comprising:

selecting the second address to be within a predetermined range of addresses that is less than a total range of addresses within the data processing system.

8. The method of claim 1 further comprising:

providing a storage device having a predetermined range of addresses for storing processing instructions to be executed by the means for executing, the processing instructions arranged in groups, each group correlated to a predetermined opcode of a program.

9. The method of claim 8 wherein the opcode is a Java bytecode.

10. The method of claim 8 further comprising:

structuring the storage device in sections of a predetermined number of instruction slots, a predetermined amount of each of the sections dedicated to storage of the instructions.

11. The method of claim 8 further comprising:

directing execution of the processing instructions from one of the sections of the storage device to an address outside of the predetermined range of addresses and subsequently redirecting program execution back to a predetermined portion of another one of the sections of the storage device.

12. The method of claim 1 further comprising:

implementing the value as a timed value provided by a timer, the timed value restricting execution of each program task implemented by the data processing system to a predetermined amount of time.

13. The method of claim 1 further comprising:

providing a storage device having a plurality of processing instructions to be executed by the means for executing, at least one of the plurality of processing instructions causing a hardware accelerator to perform a processing function.

14. The method of claim 13 wherein the change of control is signaled via a handshake signal from the means for executing to the hardware accelerator.

15. The method of claim 13 wherein the change of control is conditional based upon either the first result of the comparison or the second result of the comparison occurring.

16. The method of claim 1 further comprising:

providing a storage device having a predetermined range of addresses for storing processing instructions arranged in groups, at least two of the groups having differing numbers of processing instructions and thus differing sizes.

17. The method of claim 1 further comprising:

providing a storage device having a predetermined range of addresses for storing the plurality of processing instructions arranged in groups, at least one of the groups containing a single instruction which causes a change of flow to a separate memory resource in the data processing system for

permitting the storage device to be compressed in size, the separate memory resource containing processing instructions correlated to a predetermined opcode of a program.

18. A data processing system comprising:

a memory for storing a plurality of program instructions;

a processor coupled to the memory via a data bus for fetching the program instructions from the memory and selectively executing the plurality of program instructions; and

a storage device for storing a plurality of processing instructions to be executed by the processor, the plurality of processing instructions arranged in groups wherein each group is correlated to a predetermined one of the plurality of program instructions;

the processor executing a predetermined processing instruction at a predetermined address and selecting a jump address based upon a value by providing a first address for the jump address if a comparison of the value with a predetermined value has a first result, and by providing a second address for the jump address if the comparison of the value with the predetermined value has a second result, wherein neither the first address nor the second addresses is contiguous to the predetermined address, execution of the predetermined processing instruction always implementing a change of control in program execution.

19. The data processing system of claim 18 wherein the value is one of a count value or a timer value, the value controlling when a thread switch between program threads is required.

20. The data processing system of claim 18 wherein the processor further comprises:

an instruction register for receiving the program instructions;

an instruction decoder coupled to the instruction register for decoding the program instructions into specific opcode values;

a control circuit coupled to the instruction decoder for providing control signals in response to the specific opcode values;

address generation circuitry coupled to the control circuit for receiving the control signals and creating the jump address;

registers coupled to the address generation circuitry for storing operands in response to the control circuit; and

an arithmetic logic unit coupled to the address generation circuitry and the control circuit, the arithmetic logic unit processing the operands stored by the registers.

21. The data processing system of claim 20 wherein the control circuit further comprises a user determinable control signal that selects either a first structure of predetermined group sizes or a second structure of predetermined group sizes.

22. The data processing system of claim 20 wherein the address generation circuitry further comprises:

a multiplexer and combiner circuit coupled to the control circuit, the multiplexer and combiner circuit receiving at least one input signal that modifies the jump address.

23. A software processing instruction stored on a storage medium which, when executed by a data processor results in creation of a jump address, the jump address being based upon a value and is provided as one of either a first address or a second address, the first address being selected if a comparison of the value with a predetermined value has a first result, the second address being selected if the comparison of the value with the predetermined value has a second result, neither of the second and third addresses being contiguous to an address assignment of the software processing instruction, the software processing instruction always causing a change of control in program execution by redirecting program execution in the data processing system to the jump address.